Fast Algorithms for Minimum Cycle Basis and Minimum Homology Basis

Abhishek Rathod
Department of Mathematics, Technical University of Munich (TUM),
Boltzmannstr. 3, 85748 Garching b. München, Germany
abhishek.rathod@tum.de

Abstract

We study the problem of finding a minimum homology basis, that is, a shortest set of cycles that generates the 1-dimensional homology classes with $\mathbb{Z}_2$ coefficients in a given simplicial complex K. This problem has been extensively studied in the last few years. For general complexes, the current best deterministic algorithm, by Dey et al. [8], runs in $O(N^\omega + N^2 g)$ time, where N denotes the number of simplices in K, g denotes the rank of the 1-homology group of K, and ω denotes the exponent of matrix multiplication. In this paper, we present two conceptually simple randomized algorithms that compute a minimum homology basis of a general simplicial complex K. The first algorithm runs in $\tilde{O}(m^\omega)$ time, where m denotes the number of edges in K, whereas the second algorithm runs in $O(m^\omega + Nm^{\omega-1})$ time.

We also study the problem of finding a minimum cycle basis in an undirected graph G with n vertices and m edges. The best known algorithm for this problem runs in $O(m^\omega)$ time. Our algorithm, which has a simpler high-level description, but is slightly more expensive, runs in $\tilde{O}(m^\omega)$ time.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Algebraic topology

Keywords and phrases Computational topology, Minimum homology basis, Minimum cycle basis, Simplicial complexes, Matrix computations

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.64

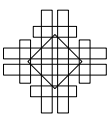
Funding This research has been supported by the DFG Collaborative Research Center SFB/TRR 109 "Discretization in Geometry and Dynamics".

Acknowledgements The author would like to thank Ulrich Bauer and Michael Lesnick for valuable discussions, and anonymous reviewers for their useful comments.

1 Introduction

Minimum cycle bases in graphs have several applications, for instance, in analysis of electrical networks, analysis of chemical and biological pathways, periodic scheduling, surface reconstruction and graph drawing. Also, algorithms from diverse application domains like electrical circuit theory and structural engineering require cycle basis computation as a preprocessing step. Cycle bases of small size offer a compact description that is advantageous from a mathematical as well as from an application viewpoint. For this reason, the problem of computing a minimum cycle basis has received a lot of attention, both in its general setting as well as in special classes of graphs such as planar graphs, sparse graphs, dense graphs, network graphs, and so on. We refer the reader to [15] for a comprehensive survey.

In topological data analysis, "holes" of different dimensions in a geometric dataset constitute "features" of the data. Algebraic topology offers a rigorous language to formalize our intuitive picture of holes in these geometric objects. More precisely, a basis for the first homology group H_1 can be taken as a representative of the one-dimensional holes in the geometric object. The advantages of using minimum homology bases are twofold: firstly, one can bring geometry in picture by assigning appropriate weights to edges, and secondly,



smaller cycles are easier to understand and analyze, especially visually. We focus solely on the bases of the first homology group since the problem of computing a shortest basis for higher homology groups with $\mathbb{Z}_2$ coefficients was shown to be NP-hard by Chen and Freedman [5].

2 Background and Preliminaries

2.1 Cycle Basis

Let $G = (V, E)$ be a connected graph. A subgraph of G which has even degree for each vertex is called a *cycle* of G. A cycle is called *elementary* if the set of edges form a connected subgraph in which each vertex has degree 2. We associate an incidence vector C, indexed on E, to each cycle, so that $C_e = 1$ if e is an edge of the cycle, and $C_e = 0$ otherwise. The set of incidence vectors of cycles forms a vector space over $\mathbb{Z}_2$, called the *cycle space* of G. It is a well-known fact that for a connected graph G, the cycle space is of dimension $|E| - |V| + 1$. Throughout, we use ν to denote the dimension of the cycle space of a graph. A basis of the cycle space, that is, a maximal linearly independent set of cycles is called a *cycle basis*.

Suppose that the edges of G have non-negative weights. Then, the weight of a cycle is the sum of the weights of its edges, and the weight of a cycle basis is the sum of the weights of the basis elements. The problem of computing a cycle basis of minimum weight is called the *minimum cycle basis* problem. Since we assume all edge weights to be non-negative, there always exists a minimum cycle basis of elementary cycles, allowing us to focus on minimum cycle basis comprising entirely of elementary cycles.

A simple cycle C is *tight* if it contains a shortest path between every pair of points in C. We denote the set of all tight cycles in the graph by $\mathcal{T}$. Tight cycles are sometimes also referred to as *isometric* cycles [1, 15]. Tight cycles play an important role in designing algorithms for minimum cycle basis, owing to the following theorem by Horton.

▶ **Theorem 1** (Horton [13]). *A minimum cycle basis $\mathcal{M}$ consists only of tight cycles.*

A key structural property about minimum cycle bases was proved by de Pina.

▶ **Theorem 2** (de Pina [7]). *Cycles $C_1 \ldots, C_\nu$ form a minimum cycle basis if there are vectors $S_1, \ldots, S_\nu$ such that for all i, $1 \leq i \leq \nu$, the following hold:*
Prefix Orthogonality: *$\langle C_j, S_i \rangle = 0$ for all $1 \leq j \leq i$.*
Non-Orthogonality: *$\langle C_i, S_i \rangle = 1$.*
Shortness: *C_i is a minimum weight cycle in $\mathcal{T}$ with $\langle C_i, S_i \rangle = 1$.*

The vectors $S_1, \ldots, S_\nu$ in Theorem 2 are called *support vectors*. The recent line of algorithmic work [1, 7, 16, 17, 18] on the minimum cycle basis problem rely on Theorem 2. In fact, these algorithms may all be seen as refinements of the algorithm by de Pina, see Algorithm 1.

Algorithm 1 De Pina's Algorithm for computing a minimum cycle basis.

```
Initialize S_i to the i-th unit vector e_i for 1 ≤ i ≤ ν
for i ← 1, ..., ν do
    Compute a minimum weight cycle C_i with ⟨C_i, S_i⟩ = 1.
    for j ← i + 1, ..., ν do
        S_j = S_j + ⟨C_i, S_j⟩ S_i
    end for
end for
Return {C_1, ..., C_ν}.
```

Algorithm 1 works by inductively maintaining a set of support vectors $\{S_i\}$ so that the conditions of Theorem 2 are satisfied when the algorithm terminates. In particular, Lines 4 and 5 of the algorithm ensure that the set of vectors S_j for $j > i$ are orthogonal to vectors $C_1, \ldots, C_i$. Updating the vectors S_j as outlined in Lines 4 and 5 of Algorithm 1 takes time $O(m^3)$ time in total. Using a divide and conquer procedure for maintaining S_j, Kavitha et al. [17] improved the cost of maintaining the support vectors to $O(m^\omega)$. See Algorithm 2.

Algorithm 2 Divide and conquer procedure for fast computation of support vectors by Kavitha et al. [17].

1: Initialize S_i to the i-th unit vector e_i for $1 \leq i \leq \nu$.
2: MinCycleBasis$(1, \nu)$.

3: **procedure** MinCycleBasis(ℓ, u)
4: **if** $\ell = u$ **then**
5: Compute a minimum weight cycle C_ℓ with $\langle C_\ell, S_\ell \rangle = 1$.
6: **else**
7: $q \leftarrow \lfloor (\ell+u)/2 \rfloor$.
8: MinCycleBasis(ℓ, q).
9: $\mathbf{C} \leftarrow [C_\ell, \ldots, C_q]$.
10: $\mathbf{W} \leftarrow (\mathbf{C}^T[S_\ell, \ldots, S_q])^{-1}\mathbf{C}^T[S_{q+1}, \ldots, S_u]$.
11: $[S_{q+1}, \ldots, S_u] \leftarrow [S_{q+1}, \ldots, S_u] + [S_\ell, \ldots, S_q]\mathbf{W}$.
12: MinCycleBasis$(q+1, u)$.
13: **end if**
14: **end procedure**
15: Return $\{C_1, \ldots, C_\nu\}$.

▶ **Lemma 3** (Lemma 5.6 in [15]). *The total number of arithmetic operations performed in lines 9 to 11 of Algorithm 2 is $O(m^\omega)$. That is, the support vectors satisfying conditions of Theorem 2 can be maintained in $O(m^\omega)$ time.*

Finally, in [1], Amaldi et al. designed an $O(m^\omega)$ time algorithm for computing a minimum cycle basis by improving the complexity of Line 5 of Algorithm 2 to $o(m^\omega)$ (from $O(m^2 n)$ in [17]), while using the $O(m^\omega)$ time divide-and-conquer template for maintaining the support vectors as presented in Algorithm 2. The $o(m^\omega)$ procedure for Line 3 is achieved by performing a Monte Carlo binary search on the set of tight cycles (sorted by weight) to find a minimum weight cycle C_i that satisfies $\langle C_i, S_i \rangle = 1$. An efficient binary search is made possible on account of the following key structural property about tight cycles.

▶ **Theorem 4** (Amaldi et al. [1]). *The total length of the tight cycles is at most $n\nu$.*

Amaldi et al. [1] also show that there exists an $O(nm)$ algorithm to compute the set of all the tight cycles of an undirected graph G. See Sections 2 and 3 of [1] for details about Amaldi et al.'s algorithm.

2.2 Matrix operations

The *column rank profile* (respectively *row rank profile*) of an $m \times n$ matrix A with rank r, is the lexicographically smallest sequence of r indices $[i_1, i_2, \ldots, i_r]$ (respectively $[j_1, j_2, \ldots, j_r]$) of linearly independent columns (respectively rows) of A. Suppose that $\{a_1, a_2, \ldots, a_n\}$ represent the columns of A. Then, following Busaryev et al. [3], we define the *earliest basis* of A as the set of columns $\mathcal{E}(A) = \{a_{i_1}, a_{i_2}, \ldots, a_{i_r}\}$.

It is well-known that classical Gaussian elimination can be used to compute rank profile in $O(nmr)$ time. The current state-of-the-art deterministic matrix rank profile algorithms run in $O(mnr^{\omega-2})$ time.

▶ **Theorem 5** ([10, 14]). *There is a deterministic $O(mnr^{\omega-2})$ time algorithm to compute the column rank profile of an $m \times n$ matrix A.*

In case of randomized algorithms, Cheung, Kwok and Lau [6] presented a breakthrough Monte Carlo algorithm for rank computation that runs in $(\mathsf{nnz}(A) + r^\omega)^{1+o(1)}$ time, where $o(1)$ in the exponent captures some missing multiplicative $\log n$ and $\log m$ factors, and $\mathsf{nnz}(A)$ denotes the number of nonzero entries in A. Equivalently, the complexity for Cheung et al.'s algorithm can also be written as $\tilde{O}(\mathsf{nnz}(A) + r^\omega)$. The notation $\tilde{O}(\cdot)$ is often used in literature to hide small polylogarithmic factors in time bounds. While the algorithm by Cheung et al. also computes r linearly independent columns of A, the columns may not correspond to the column rank profile. However, building upon Cheung et al.'s work, Storjohann and Yang established the following result.

▶ **Theorem 6** (Storjohann and Yang [19, 20, 21]). *There exists a Monte Carlo algorithm for computing row (resp. column) rank profile of a matrix A that runs in $(\mathsf{nnz}(A) + r^\omega)^{1+o(1)}$ time. The failure probability of this algorithm is $1/2$.*

Once again, the $o(1)$ in the exponent captures some missing multiplicative $\log n$ and $\log m$ factors, see [19], and hence the complexity can also be written as $\tilde{O}(\mathsf{nnz}(A) + r^\omega)$.

2.3 Homology

In this work, we restrict our attention to simplicial homology with $\mathbb{Z}_2$ coefficients. For a general introduction to algebraic topology, we refer the reader to [12]. Below we give a brief description of homology over $\mathbb{Z}_2$.

Let K be a connected simplicial complex. We will denote by $K^{(p)}$ the set of p-dimensional simplices in K, and n_p the number of p-dimensional simplices in K. Also, the p-dimensional skeleton of K will be denoted by K_p. In particular, the 1-skeleton of K (which is an undirected graph) will be denoted by K_1.

We consider formal sums of simplices with $\mathbb{Z}_2$ coefficients, that is, sums of the form $\sum_{\sigma \in K^{(p)}} a_\sigma \sigma$, where each $a_\sigma \in \{0, 1\}$. The expression $\sum_{\sigma \in K^{(p)}} a_\sigma \sigma$ is called a *p-chain*. Since chains can be added to each other, they form an Abelian group, denoted by $\mathsf{C}_p(K)$. Since we consider formal sums with coefficients coming from $\mathbb{Z}_2$, which is a field, $\mathsf{C}_p(K)$, in this case, is a vector space of dimension n_p over $\mathbb{Z}_2$. The p-simplices in K form a (natural) basis for $\mathsf{C}_p(K)$. This establishes a natural one-to-one correspondence between elements of $\mathsf{C}_p(K)$ and subsets of $K^{(p)}$. Thus, associated with each chain is an incidence vector v, indexed on $K^{(p)}$, where $v_\sigma = 1$ if σ is a simplex of v, and $v_\sigma = 0$ otherwise. The *boundary* of a p-simplex is a $(p-1)$-chain that corresponds to the set of its $(p-1)$-faces. This map can be linearly extended from p-simplices to p-chains, where the boundary of a chain is the $\mathbb{Z}_2$-sum of the boundaries of its elements. Such an extension is known as the *boundary homomorphism*, and denoted by $\partial_p : \mathsf{C}_p(K) \to \mathsf{C}_{p-1}(K)$. A chain $\zeta \in \mathsf{C}_p(K)$ is called a *p-cycle* if $\partial_p \zeta = 0$, that is, $\zeta \in \ker \partial_p$. The group of p-dimensional cycles is denoted by $\mathsf{Z}_p(K)$. As before, since we are working with $\mathbb{Z}_2$ coefficients, $\mathsf{Z}_p(K)$ is a vector space over $\mathbb{Z}_2$. A chain $\eta \in \mathsf{C}_p(K)$ is said to be a p-boundary if $\eta = \partial_{p+1} c$ for some chain $c \in \mathsf{C}_{p+1}(K)$, that is, $\eta \in \operatorname{im} \partial_{p+1}$. The group of p-dimensional boundaries is denoted by $\mathsf{B}_p(K)$. In our case, $\mathsf{B}_p(K)$ is also a vector space, and in fact a subspace of $\mathsf{C}_p(K)$.

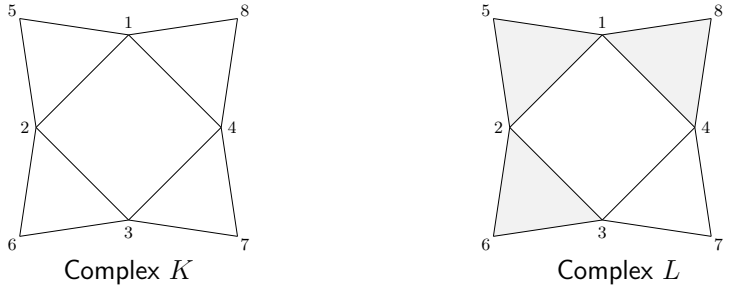


Figure 1 Consider complexes K and L in the figure above with unit weights on the edges. Since K has no 2-simplices, its 1-skeleton K_1 is identical to K itself. The set of cycles $\mathcal{C} = \{\{1,2,5\},\{1,4,8\},\{3,4,7\},\{2,3,6\},\{1,2,3,4\}\}$ constitutes a minimum cycle basis for the respective 1-skeletons K_1 and L_1 (viewed as graphs). The set $\mathcal{C}$ also constitutes a minimum homology basis for K. The set $\mathcal{C}' = \{\{1,2,3,4\},\{3,4,7\}\}$ constitutes a minimum homology basis for L.

Thus, we can consider the quotient space $\mathsf{H}_p(K) = \mathsf{Z}_p(K)/\mathsf{B}_p(K)$. The elements of the vector space $\mathsf{H}_p(K)$, known as the p-th *homology group* of K, are equivalence classes of p-cycles, where p-cycles are equivalent if their $\mathbb{Z}_2$-difference is a p-boundary. Equivalent cycles are said to be *homologous*. For a p-cycle ζ, its corresponding homology class is denoted by $[\zeta]$. Bases of $\mathsf{B}_p(K)$, $\mathsf{Z}_p(K)$ and $\mathsf{H}_p(K)$ are called *boundary bases*, *cycle bases*, and *homology bases* respectively. The dimension of the p-th homology group of K is called the p-th *Betti number* of K, denoted by $\beta_p(K)$. We are primarily interested in the first Betti number $\beta_1(K)$. For notational convenience, let $g = \beta_1(K)$, and denote the dimension of $\mathsf{B}_1(K)$ by b.

Using the natural bases for $\mathsf{C}_p(K)$ and $\mathsf{C}_{p-1}(K)$, the matrix $[\partial_p\sigma_1\ \partial_p\sigma_2 \cdots \partial_p\sigma_{n_p}]$ whose column vectors are boundaries of p-simplices is called the p-th boundary matrix. Abusing notation, we denote the p-th boundary matrix by ∂_p. For the rest of the paper, we use n, m and N to denote the number of vertices, edges and simplices in the complex respectively.

A set of p-cycles $\{\zeta_1, \ldots, \zeta_g\}$ is called a *homology cycle basis* if the set of classes $\{[\zeta_1], \ldots, [\zeta_g]\}$ forms a homology basis. For brevity, we abuse notation by using the term "homology basis" for $\{\zeta_1, \ldots, \zeta_g\}$. Assigning non-negative weights to the edges of K, the weight of a cycle is the sum of the weights of its edges, and the weight of a homology basis is the sum of the weights of the basis elements. The problem of computing a minimum weight basis of $\mathsf{H}_1(K)$ is called the *minimum homology basis* problem. Note that, when the input simplicial complex is a graph, the notions of homology basis and cycle basis coincide. Please refer to Figure 1 for an example.

For the special case when the input complex is a surface, Erickson and Whittlesey [11] gave a $O(N^2 \log N + gN^2 + g^3N)$-time algorithm. Recently, Borradaile et al. [2] gave an improved deterministic algorithm that runs in $O((h+c)^3 n \log n + m)$ where c denotes the number of boundary components, and h denotes the genus of the surface. For small values of c and h, the algorithm runs in nearly linear time.

For general complexes, Dey et al. [9] and Chen and Freedman [4] generalized the results by Erickson and Whittlesey [11] to arbitrary complexes. Subsequently, introducing the technique of annotations, Busaryev et al. [3] improved the complexity to $O(N^\omega + N^2 g^{\omega-1})$. More recently, Dey et al. [8] designed an $O(N^\omega + N^2 g)$ time algorithm by adapting the divide and conquer algorithm for computing a minimum cycle basis of Kavitha et al. [17] for the purpose of computing a minimum homology basis. Dey et al. also designed a randomized 2-approximation algorithm for the same problem that runs in $O(N^\omega \sqrt{N \log N})$ expected time.

3 An algorithm for computing minimum cycle basis

Given a graph $G = (V, E)$, let $\{C_1, \ldots, C_{|\mathcal{T}|}\}$ be the list of tight cycles in G sorted by weight, and let $\mathbf{M}_{\mathcal{T}}(G) = [C_1\, C_2 \ldots\, C_{|\mathcal{T}|}]$ be the matrix formed with cycles C_i as its columns. Using Theorem 4, since the total length of tight cycles is at most $n\nu$, and since each tight cycle consists of at least three edges, we have that $|\mathcal{T}| \leq \frac{n\nu}{3}$. Also, the rank of $\mathbf{M}_{\mathcal{T}}(G)$ is ν and $\mathbf{M}_{\mathcal{T}}(G)$ is a sparse matrix with $\mathsf{nnz}(\mathbf{M}_{\mathcal{T}}(G))$ bounded by $n\nu$. This sparsity is implicitly used in the design of the Monte Carlo binary search algorithm for computing minimum cycle basis, as described in [1]. We now present a simple and fast algorithm for minimum cycle basis that exploits the sparsity and the low rank of $\mathbf{M}_{\mathcal{T}}(G)$ more directly.

Algorithm 3 Algorithm for minimum cycle basis.

1: Compute the sorted list of tight cycles in G, and assemble the matrix $\mathbf{M}_{\mathcal{T}}(G)$.
2: Compute the column rank profile $[i_1, i_2, \ldots, i_\nu]$ of $\mathbf{M}_{\mathcal{T}}(G)$ using Storjohann and Yang's algorithm described in [20].
3: RETURN $\mathcal{E}(\mathbf{M}_{\mathcal{T}}(G))$.

▶ **Theorem 7.** *There is a Monte Carlo algorithm that computes the minimum cycle basis in $\tilde{O}(m^\omega)$ time, with failure probability at most $1/2$.*

Proof. The correctness of the algorithm follows immediately from Theorem 1. For instance, if $\mathcal{E}(\mathbf{M}_{\mathcal{T}}(G))$ is not a minimum cycle basis, then let k be the smallest integer such that the k-th smallest cycle in a minimum cycle basis contained in $\mathbf{M}_{\mathcal{T}}(G)$ is smaller than the k-th smallest cycle in $\mathcal{E}(\mathbf{M}_{\mathcal{T}}(G))$. Since the columns in $\mathbf{M}_{\mathcal{T}}(G)$ are sorted by weight, the existence of such a k contradicts the fact that $\mathcal{E}(\mathbf{M}_{\mathcal{T}}(G))$ is the earliest basis of $\mathbf{M}_{\mathcal{T}}(G)$.

The list of tight cycles in G can be computed in $O(nm)$ time using the algorithm described in Section 2 of [1]. Hence, Step 1 of Algorithm 3 takes $O(nm \log(nm))$ time (which in turn is same as $O(nm \log n)$ time). Moreover, using Theorem 6, the complexity of Step 2 is bounded by $\tilde{O}(n\nu + \nu^\omega)$. Since $n, \nu < m$, the complexity of Algorithm 3 is bounded by $\tilde{O}(m^\omega)$. Using Theorem 6, the failure probability of the algorithm is at most $1/2$. ◀

4 Minimum homology basis, minimum cycle basis and tight cycles

To begin with, note that since every graph is a 1-dimensional simplicial complex, the minimum cycle basis problem is a restriction of the minimum homology basis problem to instances (simplicial complexes) that have no 2-simplices. In this section, we refine this observation by deriving a closer relation between the two problems.

We assume that we are provided a complex K in which all edges are assigned non-negative weights. Given a non-negative weight $w(\sigma)$ for each edge σ, we define the weight of a cycle z as the sum of the weights of the edges, $w(z) = \sum_{\sigma \in z} w(\sigma)$. Let $\mathcal{B} = \{\eta_1, \ldots, \eta_b\}$ be a basis for the boundary vector space $\mathsf{B}_1(K)$ indexed so that $w(\eta_i) \leq w(\eta_{i+1})$, $1 \leq i < b$ (with ties broken arbitrarily). Also, let $\mathcal{H} = \{\zeta_1, \ldots, \zeta_g\}$ be a minimum homology basis of K indexed so that $w(\zeta_i) \leq w(\zeta_{i+1})$, $1 \leq i < g$ (with ties broken arbitrarily). Then, the set $\mathcal{C} = \{\eta_1, \ldots, \eta_b, \zeta_1, \ldots, \zeta_g\}$ is a cycle basis for K_1. Let $\mathcal{M}$ be a minimum cycle basis of K_1. Every element $C \in \mathcal{M}$ is homologous to a cycle $\sum_{i=1}^{g} a_i \zeta_i$ where $a_i \in \{0, 1\}$ for each i. Then, for some fixed integers p and q, $\mathcal{M} = \{B_1, \ldots, B_q, C_1, \ldots, C_p\}$ is indexed so that the elements $B_1, \ldots, B_q$ are null-homologous and the elements $C_1, \ldots, C_p$ are non-bounding cycles. Also, we have $w(B_j) \leq w(B_{j+1})$ for $1 \leq j < q$ (with ties broken arbitrarily), and $w(C_j) \leq w(C_{j+1})$ for $1 \leq j < p$ (with ties broken arbitrarily).

▶ **Lemma 8.**
1. *For every minimum homology basis,* $w(\zeta_1) = w(C_1)$.
2. *There exists a minimum homology basis* $\overline{\mathcal{H}}$ *with* ζ_1 *homologous to* C_1.

Proof. Suppose there exists a minimum homology basis with $w(\zeta_1) < w(C_1)$. Let $\zeta_1 = \sum_{i=1}^{p} a_i C_i + \sum_{j=1}^{q} b_j B_j$, where $a_i \in \{0,1\}$ for each i and $b_j \in \{0,1\}$ for each j. Since ζ_1 is a non-bounding cycle, there exists at least one i with $a_i = 1$. Let $\ell \in [1,p]$ be the largest index in the above equation with $a_\ell = 1$. Rewriting the equation, we obtain $C_\ell = \sum_{i=1}^{\ell-1} a_i C_i + \sum_{j=1}^{q} b_j B_j + \zeta_1$. Since $w(\zeta_1) < w(C_1)$ by assumption, we have $w(\zeta_1) < w(C_\ell)$ because $w(C_\ell) \geq w(C_1)$ by indexing of $\mathcal{M}$. It follows that the basis obtained by exchanging C_ℓ for ζ_1, that is, $\{B_1, \ldots, B_q, \zeta_1, C_1, \ldots, C_{\ell-1}, C_{\ell+1}, \ldots, C_p\}$ gives a smaller cycle basis than the minimum one, a contradiction.

Now, suppose there exists a minimum homology basis with $w(\zeta_1) > w(C_1)$. Let $C_1 = \sum_{i=1}^{g} a_i \zeta_i + \sum_{j=1}^{b} b_j \eta_j$. As before, since C_1 is not null-homologous, there exists at least one i with $a_i = 1$. Let $\ell \in [1,g]$ be the largest index in the above equation with $a_\ell = 1$. Then, $\zeta_\ell = \sum_{i=1}^{\ell-1} a_i \zeta_i + \sum_{j=1}^{b} b_j \eta_j + C_1$. Note that $w(\zeta_\ell) \geq w(\zeta_1)$ because of the indexing, and $w(\zeta_1) > w(C_1)$ by assumption. Therefore, the set $\{C_1, \zeta_1, \ldots, \zeta_{\ell-1}, \zeta_{\ell+1}, \ldots, \zeta_p\}$ obtained by exchanging ζ_ℓ for C_1 gives a smaller homology basis than the minimum one, a contradiction. This proves the first part of the lemma.

From the first part of the lemma, we have $w(\zeta_1) = w(C_1)$ for every minimum homology basis. Let $\mathcal{H}$ be an arbitrary minimum homology basis. Then, if C_1 is not homologous to $\zeta_1 \in \mathcal{H}$, by using basis exchange we can obtain $\overline{\mathcal{H}} = \{C_1, \zeta_1, \ldots, \zeta_{\ell-1}, \zeta_{\ell+1}, \ldots, \zeta_p\}$, which is the minimum homology basis with its first element homologous to C_1, and having the same weight as $w(C_1)$, proving the claim. ◀

We now prove a theorem which allows us to harness fast algorithms for minimum cycle basis in service of improving time complexity of algorithms for minimum homology basis.

▶ **Theorem 9.** *Given a simplicial complex* K*, and a minimum cycle basis* $\mathcal{M} = \{B_1, \ldots, B_q, C_1, \ldots, C_p\}$ *of* K_1*, there exists a minimum homology basis* $\overline{\mathcal{H}}$ *of* K*, and a set* $\{C_{i_1}, \ldots, C_{i_g}\} \subset \{C_1, \ldots, C_p\} \subset \mathcal{M}$ *such that, for every* $k \in [1,g]$*, we have* C_{i_k} *homologous to a cycle spanned by* $\zeta_1, \ldots, \zeta_k$*, and* $w(C_{i_k}) = w(\zeta_k)$*. Moreover,* $i_1 = 1$*, and* i_k *for* $k > 1$ *is the smallest index for which* C_{i_k} *is not homologous to any cycle spanned by* $\{C_{i_1}, \ldots, C_{i_{k-1}}\}$*. In particular, the set* $\{C_{i_1}, \ldots, C_{i_g}\} \subset \mathcal{M}$ *constitutes a minimum homology basis of* K*.*

Proof. The key argument is essentially the same as for the proof of Lemma 8. Nonetheless, we present it here for the sake of completeness. We shall prove the claim by induction. Lemma 8 covers the base case. By induction hypothesis, there is an integer k, and a minimum homology basis $\mathcal{H} = \{\zeta_1, \ldots, \zeta_g\}$, for which, vectors $\{C_{i_1}, \ldots, C_{i_k}\} \subseteq \{C_1, \ldots, C_p\}$ are such that, for every $j \in [1,k]$, we have C_{i_j} homologous to a cycle spanned by $\zeta_1, \ldots, \zeta_j$, and $w(C_{i_j}) = w(\zeta_j)$. Let i_{k+1} be the smallest index for which $C_{i_{k+1}} \in \{C_1, \ldots, C_p\}$ is not homologous to any cycle spanned by $\{C_{i_1}, \ldots, C_{i_k}\}$.

Suppose that $w(\zeta_{k+1}) < w(C_{i_{k+1}})$. Let $\zeta_{k+1} = \sum_{i=1}^{p} a_i C_i + \sum_{j=1}^{q} b_j B_j$. Let $\ell \in [1,p]$ be the largest index in the above equation with $a_\ell = 1$. Then, $C_\ell = \sum_{i=1}^{\ell-1} a_i C_i + \sum_{j=1}^{q} b_j B_j + \zeta_{k+1}$. From the induction hypothesis, we can infer that $\ell \geq i_{k+1}$, and hence $w(C_\ell) \geq w(C_{i_{k+1}})$ by indexing of $\mathcal{M}$. Thus, if $w(\zeta_{k+1}) < w(C_{i_{k+1}})$, then we have $w(\zeta_{k+1}) < w(C_\ell)$. It follows that, $\{B_1, \ldots, B_q, \zeta_{k+1}, C_1, \ldots, C_{\ell-1}, C_{\ell+1}, \ldots, C_p\}$ obtained by exchanging C_ℓ for ζ_{k+1} gives a smaller cycle basis than the minimum one, contradicting the minimality of $\mathcal{H}$.

Now, suppose that $w(\zeta_{k+1}) > w(C_{i_{k+1}})$. Let $C_{i_{k+1}} = \sum_{i=1}^{g} a_i \zeta_i + \sum_{j=1}^{b} b_j \eta_j$. Let $\ell \in [1,g]$ be the largest index in the above equation with $a_\ell = 1$. Rewriting the equation, we obtain $\zeta_\ell = \sum_{i=1}^{\ell-1} a_i \zeta_i + \sum_{j=1}^{b} b_j \eta_j + C_{i_{k+1}}$. Again, using the induction hypothesis, $\ell \geq k+1$, and

hence, $w(\zeta_\ell) \geq w(\zeta_{k+1})$ because of the indexing. Since we have assumed $w(\zeta_{k+1}) > w(C_{i_{k+1}})$, this gives us $w(\zeta_\ell) > w(C_{i_{k+1}})$. Hence, the set $\{C_{i_{k+1}}, \zeta_1, \ldots, \zeta_{\ell-1}, \zeta_{\ell+1}, \ldots, \zeta_p\}$ obtained by exchanging ζ_ℓ for $C_{i_{k+1}}$ gives a smaller homology basis than the minimum one, contradicting the minimality of $\mathcal{H}$.

From the first part of the proof, we have established that $w(C_{i_{k+1}}) = w(\zeta_{k+1})$. So, if $C_{i_{k+1}}$ is not homologous to $\zeta_{k+1} \in \mathcal{H}$ and $w(\zeta_{k+1}) = w(C_{i_{k+1}})$, then $\overline{\mathcal{H}} = \{C_{i_{k+1}}, \zeta_1, \ldots, \zeta_{\ell-1}, \zeta_{\ell+1}, \ldots, \zeta_p\}$ obtained by exchanging ζ_ℓ for $C_{i_{k+1}}$ is the desired minimum homology basis, proving the induction claim. ◀

Previously, it was known from Erickson and Whittlesey [11] that $\mathcal{H}$ is contained in $\mathcal{T}$.

▶ **Theorem 10** (Erickson and Whittlesey [11])**.** *With non-negative weights, every cycle in a shortest basis of $\mathsf{H}_1(K)$ is tight. That is, if $\mathcal{H}$ is any minimum homology basis of K, then $\mathcal{H} \subset \mathcal{T}$.*

Using Theorems 1 and 9, we can refine the above observation.

▶ **Corollary 11.** *Let $\mathcal{T}$ denote the set of tight cycles of K_1, and let $\mathcal{M}$ be a minimum cycle basis of K_1. Then, there exists a minimum homology basis $\mathcal{H}$ of K such that $\mathcal{H} \subset \mathcal{M} \subset \mathcal{T}$.*

5 Algorithms for minimum homology basis

To begin with, note that since $\mathsf{C}_p(K), \mathsf{Z}_p(K), \mathsf{B}_p(K)$ and $\mathsf{H}_p(K)$ are vector spaces, the problem of computing a minimum homology basis can be couched in terms of matrix operations.

Given a complex K, let $\{C_1, \ldots, C_{|\mathcal{T}|}\}$ be the list of tight cycles in K_1 sorted by weight, and let $\mathbf{M}_{\mathcal{T}}(K_1) = [C_1\, C_2 \ldots\, C_{|\mathcal{T}|}]$ be the matrix formed with cycles C_i as its columns. Then, the matrix $\hat{\mathbf{Z}} = [\partial_2 \mid \mathbf{M}_{\mathcal{T}}(K_1)]$ has $O(N + n\nu)$ columns and $O(N + n\nu)$ non-zero entries since $\mathbf{M}_{\mathcal{T}}(K_1)$ has $O(n\nu)$ columns and $O(n\nu)$ non-zero entries by Theorem 4, and ∂_2 has $O(N)$ columns and $O(N)$ non-zero entries. Since $\hat{\mathbf{Z}}$ has m rows, the rank of $\hat{\mathbf{Z}}$ is bounded by m. This immediately suggests an algorithm for computing minimum homology basis analogous to Algorithm 3.

Algorithm 4 Algorithm for minimum homology basis.

1: Compute the sorted list of tight cycles in $\mathbf{M}_{\mathcal{T}}(K_1)$, and assemble matrix $\hat{\mathbf{Z}}$.
2: Compute the column rank profile $[j_1, j_2, \ldots, j_b, i_1, i_2, \ldots, i_g]$ of $\hat{\mathbf{Z}}$ using Storjohann and Yang's algorithm [20], where columns $\{\hat{\mathbf{Z}}_{j_k}\}$ and $\{\hat{\mathbf{Z}}_{i_\ell}\}$ are linearly independent columns of ∂_2 and $\mathbf{M}_{\mathcal{T}}(K_1)$ respectively.
3: Return Columns $\{\hat{\mathbf{Z}}_{i_1}, \hat{\mathbf{Z}}_{i_2}, \ldots, \hat{\mathbf{Z}}_{i_g}\}$.

▶ **Theorem 12.** *Algorithm 4 is a Monte Carlo algorithm for computing a minimum homology basis that runs in $\tilde{O}(m^\omega)$ time with failure probability at most $\frac{1}{2}$.*

Proof. The correctness of the algorithm is an immediate consequence of Theorem 9 since, by definition, i_k is the smallest index for which $\hat{\mathbf{Z}}_{i_k}$ is not homologous to any cycle spanned by $\{\hat{\mathbf{Z}}_{i_1}, \ldots, \hat{\mathbf{Z}}_{i_{k-1}}\}$.

The list of tight cycles in G can be computed in $O(nm)$ time using the algorithm described in Section 2 of [1]. Hence, Step 1 of Algorithm 4 takes $O(nm \log n)$ time. Moreover, using Theorem 6, the complexity of Step 2 is bounded by $\tilde{O}(N + n\nu + m^\omega)$, which is the same as $\tilde{O}(m^\omega)$ since N and $n\nu$ are both in $\tilde{O}(m^\omega)$, and the failure probability is at most $1/2$. ◀

When the number of 2-simplices in complex K is significantly smaller than the number of edges, the complexity for minimum homology can be slightly improved by decoupling the minimum homology basis computation from the minimum cycle basis computation, as illustrated in Algorithm 5.

Algorithm 5 Algorithm for minimum homology basis.

1: Compute a minimum cycle basis $\mathcal{M}$ of K_1 using the Monte Carlo algorithm by Amaldi et al. [1]. Let $\mathbf{B}_{\mathcal{M}}$ be the matrix whose columns are cycle vectors in $\mathcal{M}$ sorted by weight.
2: Assemble the matrix $\tilde{\mathbf{Z}} = [\partial_2 \mid \mathbf{B}_{\mathcal{M}}]$.
3: Compute the column rank profile $[j_1, j_2, \ldots, j_b, i_1, i_2, \ldots, i_g]$ of $\tilde{\mathbf{Z}}$ using the deterministic algorithm by Jeannerod et al. [14], where columns $\{\tilde{\mathbf{Z}}_{j_k}\}$ and $\{\tilde{\mathbf{Z}}_{i_\ell}\}$ are linearly independent columns of ∂_2 and $\mathbf{B}_{\mathcal{M}}$ respectively.
4: RETURN Columns $\{\tilde{\mathbf{Z}}_{i_1}, \tilde{\mathbf{Z}}_{i_2}, \ldots, \tilde{\mathbf{Z}}_{i_g}\}$.

▶ **Theorem 13.** *Minimum homology basis can be computed in $O(m^\omega + Nm^{\omega-1})$ time using the Monte Carlo algorithm described in Algorithm 5. The algorithm fails with probability at most $\nu \log(nm)\, 2^{-k}$, where $k = m^{0.1}$.*

Proof. As in Theorem 12, the correctness of the algorithm is an immediate consequence of Theorem 9. The algorithm fails only when Step 1 returns an incorrect answer, the probability of which is as low as $\nu \log(nm)\, 2^{-k}$, where $k = m^{0.1}$, see Theorem 3.2 of [1].

The minimum cycle basis algorithm by Amaldi et al. [1] runs in $O(m^\omega)$ time (assuming the current exponent of matrix multiplication $\omega > 2$). Furthermore, using Theorem 5, the complexity of Line 3 is bounded by $O(Nm^{\omega-1})$. So, the overall complexity of the algorithm is $O(m^\omega + Nm^{\omega-1})$. ◀

Note that in Line 3 of Algorithm 5, it is possible to replace the deterministic algorithm by Jeannerod et al. [14] with the Monte Carlo algorithm by Storjohann and Yang's algorithm [20]. In that case, the complexity of the algorithm will once again be $\tilde{O}(m^\omega)$, and the failure probability will be at most $1 - \frac{1}{2}(1 - \nu \log(nm) 2^{-k})$.

We would like to point out that the complexities of Algorithm 4 and Algorithm 5 are, *in general*, not comparable. For instance, for families of complexes with $N^{1-\epsilon} = \omega(m)$, for some $\epsilon > 0$, Algorithm 4 is faster than Algorithm 5. However, for families of complexes with $N = o(m)$, Algorithm 5 is faster than Algorithm 4. Moreover, for families of complexes with $g = \Theta(N)$, where, as before, g denotes the rank of $\mathsf{H}_1(K)$, Algorithms 4 and 5 are both faster than Dey et al.'s algorithm [8] (which runs in $O(N^\omega + N^2 g)$ time).

6 Discussion

In this paper, we show that questions about minimum cycle basis and minimum homology basis can be naturally recast into the problem of computing rank profiles of matrices, leading to fast algorithms with simple and elegant high-level descriptions. The column rank profile (or the earliest basis) of a matrix has previously been used to compute the minimum homology basis of a simplicial complex [3, 8]. Such a greedy approach that picks, at each step, an independent cycle of the smallest index, works because of the matroid structure of homology bases and cycle bases. What's novel about our approach is that we point out that, for both problems, independence can be efficiently checked owing to the sparsity of the matrices comprising of candidate cycles.

It is also worth noting that for the algorithms presented in this paper, the simplicity of high-level description doesn't translate to simple algorithms that can be easily implemented because the black-box subroutines employed by these algorithms are fairly complex.

Maintenance of support vectors has served as a key ingredient in designing algorithms for minimum cycle basis since de Pina. Our algorithm, however, does not explicitly maintain support vectors, and in that sense, is somewhat conceptually different from the recent algorithms for computing minimum cycle bases.

References

1 Edoardo Amaldi, Claudio Iuliano, Tomasz Jurkiewicz, Kurt Mehlhorn, and Romeo Rizzi. Breaking the $O(m^2n)$ barrier for minimum cycle bases. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009*, pages 301–312, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

2 Glencora Borradaile, Erin Wolf Chambers, Kyle Fox, and Amir Nayyeri. Minimum cycle and homology bases of surface-embedded graphs. *JoCG*, 8(2):58–79, 2017. doi:10.20382/jocg.v8i2a4.

3 Oleksiy Busaryev, Sergio Cabello, Chao Chen, Tamal K. Dey, and Yusu Wang. Annotating simplices with a homology basis and its applications. In Fedor V. Fomin and Petteri Kaski, editors, *Algorithm Theory – SWAT 2012*, pages 189–200, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

4 Chao Chen and Daniel Freedman. Measuring and computing natural generators for homology groups. *Comput. Geom. Theory Appl.*, 43(2):169–181, February 2010. doi:10.1016/j.comgeo.2009.06.004.

5 Chao Chen and Daniel Freedman. Hardness results for homology localization. *Discrete & Computational Geometry*, 45(3):425–448, April 2011. doi:10.1007/s00454-010-9322-8.

6 Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. *J. ACM*, 60(5):31:1–31:25, October 2013. doi:10.1145/2528404.

7 José Coelho de Pina. *Applications of shortest path methods.* PhD thesis, Universiteit van Amsterdam, 1995.

8 Tamal K. Dey, Tianqi Li, and Yusu Wang. Efficient algorithms for computing a minimal homology basis. In Michael A. Bender, Martín Farach-Colton, and Miguel A. Mosteiro, editors, *LATIN 2018: Theoretical Informatics*, pages 376–398, Cham, 2018. Springer International Publishing.

9 Tamal K. Dey, Jian Sun, and Yusu Wang. Approximating loops in a shortest homology basis from point data. In *Proceedings of the Twenty-sixth Annual Symposium on Computational Geometry*, SoCG '10, pages 166–175, New York, NY, USA, 2010. ACM. doi:10.1145/1810959.1810989.

10 Jean-Guillaume Dumas, Clément Pernet, and Ziad Sultan. Simultaneous computation of the row and column rank profiles. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, ISSAC '13, pages 181–188, New York, NY, USA, 2013. ACM. doi:10.1145/2465506.2465517.

11 Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1038–1046, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

12 Allen Hatcher. *Algebraic topology.* Cambridge University Press, Cambridge, 2002.

13 J. D. Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM J. Comput.*, 16(2):358–366, April 1987. doi:10.1137/0216026.

14 Claude-Pierre Jeannerod, Clément Pernet, and Arne Storjohann. Rank-profile revealing gaussian elimination and the cup matrix decomposition. *Journal of Symbolic Computation*, 56:46–68, 2013. doi:10.1016/j.jsc.2013.04.004.

15 Telikepalli Kavitha, Christian Liebchen, Kurt Mehlhorn, Dimitrios Michail, Romeo Rizzi, Torsten Ueckerdt, and Katharina A. Zweig. Cycle bases in graphs characterization, algorithms, complexity, and applications. *Computer Science Review*, 3(4):199–243, 2009. doi:10.1016/j.cosrev.2009.08.001.

16 Telikepalli Kavitha, Kurt Mehlhorn, and Dimitrios Michail. New approximation algorithms for minimum cycle bases of graphs. *Algorithmica*, 59(4):471–488, April 2011. doi:10.1007/s00453-009-9313-4.

17 Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna Paluch. A faster algorithm for minimum cycle basis of graphs. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming*, pages 846–857, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

18 Kurt Mehlhorn and Dimitrios Michail. Minimum cycle bases: Faster and simpler. *ACM Trans. Algorithms*, 6(1):8:1–8:13, December 2009. doi:10.1145/1644015.1644023.

19 Arne Storjohann and Shiyun Yang. Linear independence oracles and applications to rectangular and low rank linear systems. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 381–388, New York, NY, USA, 2014. ACM. doi:10.1145/2608628.2608673.

20 Arne Storjohann and Shiyun Yang. A relaxed algorithm for online matrix inversion. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '15, pages 339–346, New York, NY, USA, 2015. ACM. doi:10.1145/2755996.2756672.

21 Shiyun Yang. Algorithms for fast linear system solving and rank profile computation. Master's thesis, University of Waterloo, 2014.